Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-29. (Canceled)

30.   (Currently Amended) A system for parsing a web-document based on elements, which is provided to an application of a handheld terminal when the system calls the web-document to provide it to the handheld terminal, comprising:

a word parser ~~for separating~~ that separates a token on the basis of markup and non-markup by referring to a token table for all markup data necessary for a kind of document to be supported, wherein a same string of the web-document provided to the terminal has a different token according to whether it is a markup or a non-markup; and

a syntax parser ~~for parsing~~ that parses a contents model on the basis of a document type definition (DTD) of each document, ~~parsing~~ parses each syntax on the basis of the result of parsing the contents model, and ~~generating~~ generates a tree-based object on the basis of graphic user interface (GUI) of the terminal.

31.   (Currently Amended)   The system of claim 30, wherein the word parser comprises:

a comment parser ~~for processing~~ that processes a comment and a space;

a markup start parser ~~for recognizing~~ that recognizes a markup start tag and ~~generating~~ generates a token;

an attribute parser ~~for parsing~~ that parses an attribute and ~~generating~~ generates a token; and

a parsed character data analyzer ~~for analyzing~~ that analyzes parsed character data and ~~generating~~ generates a token, wherein the attribute parser is configured to recognize a name of an attribute or to recognize a value of the attribute.

32.   (Currently Amended)   The system of claim 30, wherein the syntax parser comprises:

an XML verifier ~~for verifying~~ that verifies whether a corresponding document is composed suitable for each DTD on the basis of the token generated by the word parser; and

a terminal GUI-based object generator ~~for matching~~ that matches the analyzed markup and a GUI of the terminal.

33.   (Previously Presented) The system of claim 30, wherein the parsing system integrally parses a web-document composed on the basis of any one of SGML and XML related to HTML, XHTML, mHTML, cHTML, WML and HDML.

34.   (Currently Amended) The system of claim 30, wherein the parsing system can be applied to any handheld terminal and select a̲ kind of an element to be parsed according to a̲ specification of each of the terminals.

35.   (Currently Amended) A method for parsing a called web-document of a web-server, the method comprising ~~the steps of~~:

(a) reading a token from the web-document and parsing the token;

(b) if the token is not a defined start tag or if the token is a comment or a space as result of the ~~step (a)~~reading, ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

(c) parsing the attribute of the element from the token, storing GUI-related information of the element, and parsing contents of the element;

(d) as the result of the ~~step (c)~~ parsing, if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears; and

~~(e)~~ in the case __that__ the contents of the element are not the parsed character data,

if the end tag corresponding to the __defined__ start tag ~~defined~~ appears, terminating, and if the end

tag __corresponding to the defined start tag__ does not appear, ignoring and returning.

36.   (Currently Amended)   The method of claim 35, wherein the ~~step (e)__ parsing

comprises ~~the steps of~~:

if the read token does not include a defined start tag, reading the data

continuously until the end tag appears, ~~thereby__ and if the end tag corresponding to the defined

start tag does not appear,__ ignoring the token; and

reading a new token.

37.   (Currently Amended) A recording medium for storing a program for parsing a

called web-document of a web-server, the recording medium being read by a computer, the

program comprising the functions of:

~~(a)~~ reading a token from the web-document and parsing the token;

~~(b)~~ if the token is not a defined start tag or if the token is a comment or a space as

a result of the ~~function (a)__ reading, ignoring the token, and when the defined start tag is read,

parsing an attribute of an element from the token;

(c) parsing the attribute of the element from the token, storing GUI-related information of the element, and parsing contents of the element;

(d) if the contents of the element are parsed character data as result of the function (c) parsing, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears; and

(e) in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag defined appears, terminating, and if the end tag does not appear, ignoring and returning.

38.   (Currently Amended) A system for parsing a web-document based on elements to provide contents thereof to a handheld terminal, comprising:

a word parser for extracting that extracts and separating separates tokens representing the web-document supplied regardless of a kind of a markup language used to compose the web-document by referring to a token table, wherein a same string of the web-document provided to the terminal has a different token according to whether it is a markup or a non-markup; and

a syntax parser for parsing that parses syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generating generates an object on the basis of GUI of the terminal.

39.   (Previously Presented)   The system of claim 38, wherein the token table comprises:

tokens defined in an XML document;

keywords defined in DTD for all documents provided to the handheld terminal; and

a list of elements which can be supported by each terminal.

40.   (Currently Amended)   The system of claim 38, wherein the word parser comprises:

a comment parser ~~for recognizing~~ that recognizes a comment or a space and ~~generating~~ generates a token;

a markup start parser ~~for recognizing~~ that recognizes a markup start tag and ~~generating~~ generates a token;

an attribute parser ~~for parsing~~ that parses an attribute and ~~generating~~ generates a token; and

a parsed character data analyzer ~~for analyzing~~ that analyzes parsed character data and ~~generating~~ generates a token.

41.   (Previously Presented)   The system of claim 38, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

42.   (Currently Amended) The system of claim 38, wherein the contents model means a hierarchy of elements and an attribute list, and is defined in a_DTD for all documents provided to the handheld terminal.

43.   (Currently Amended)   The system of claim 38, wherein the syntax parser comprises:

an XML verifier ~~for verifying~~ that verifies whether a web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator ~~for matching~~ that matches the parsed syntax and a GUI of the terminal.

44.   (Currently Amended) A system for parsing web-document based on elements, comprising:

a token table comprising tokens defined in an XML document, keywords defined in a document type definition (DTD) for documents provided to a handheld terminal, and a list of elements, which can be supported by each terminal;

a word parser ~~for extracting~~ that extracts and ~~separating~~ separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute;

a contents model determined by DTDs for the documents provided to the terminal that includes a hierarchy of elements and an attribute list; and

a syntax parser ~~for parsing~~ that parses syntax for the tokens extracted and separated by the word parser on the basis of the contents model, and ~~generating~~ generates an object on the basis of GUI of the terminal through the parsed syntax.

45.   (Currently Amended) The system of claim 44, the word parser comprises:

a comment parser ~~for recognizing~~ that recognizes a comment or a space and ~~generating~~ generates a token;

a markup start parser ~~for recognizing~~ that recognizes a markup start tag and ~~generating~~ generates a token;

9

~~an~~ the attribute parser ~~for parsing~~ that parses an attribute and ~~generating~~ generates a token; and

a parsed character data analyzer ~~for analyzing~~ that analyzes parsed character data and ~~generating~~ generates a token.

46.   (Previously Presented)   The system of claim 44, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

47.   (Currently Amended)   The system of claim 44, wherein the syntax parser comprises:

an XML verifier ~~for verifying~~ that verifies whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator ~~for matching~~ that matches the parsed syntax and a GUI of the terminal.

48.   (Currently Amended)   A handheld terminal, comprising:

an integral parser ~~for parsing~~ that parses a web-document composed of a predetermined markup language supplied from a web-server;

a memory ~~for storing~~ that stores information parsed by the integral parser; and

an application program using information extracted from the integral parser, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute.

49.   (Currently Amended)   A handheld terminal comprising an antenna, a CPU, a peripheral circuit, a vocoder, a memory, and an audio codec, wherein the memory ~~comprising~~ comprises:

an integral parser ~~for calling~~ that calls a web-document supplied from a web-server regardless of a markup language used to compose the web-document and ~~parsing~~ parses the web-document on the basis of elements, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and

wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute; and

an application program using information extracted from the integral parser.

50.   (Currently Amended) The handheld terminal of claim 48, wherein the integral parser comprises:

a token table comprising tokens defined in an XML document, keywords defined in a DTD for all documents provided to the handheld terminal, and a list of elements, which can be supported by each of the handheld terminals;

a the word parser ~~for extracting~~ that extracts and ~~separating~~ separates all tokens of the document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to a token table;

a contents model defined in the DTD for all documents provided to the terminal and meaning a hierarchy of the elements and an attribute list; and

a syntax parser ~~for parsing~~ that parses syntax for the tokens extracted and separated by the word parser on the basis of contents model, and ~~generating~~ generates an object on the basis of GUI of the terminal through the parsed syntax.

51.   (Currently Amended)   The system of claim 50, <u>wherein</u> the word parser

comprises:

a comment parser ~~for recognizing~~ that recognizes a comment or a space and

~~generating~~ generates a token;

a markup start parser ~~for recognizing~~ that recognizes a markup start tag and

~~generating~~ generates a token;

an attribute parser ~~for parsing~~ that parses an attribute and ~~generating~~ generates a

token; and

a parsed character data analyzer ~~for analyzing~~ that analyzes parsed character data

and ~~generating~~ generates a token.

52.   (Previously Presented)   The system of claim 50, wherein the word parser

comprises a token generator and an XML well-formedness verifier, receives the supplied

document character by character, recognizes a token of the document on the basis of the token

table, and extracts the token by using the token generator and the XML well-formedness verifier.

53.   (Currently Amended)   The system of claim 50, wherein the syntax parser

comprises:

an XML verifier ~~for verifying~~ that verifies whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator ~~for matching~~ that matches the parsed syntax and a GUI of the terminal.

54.   (Previously Presented)   The handheld terminal of claim 48, wherein the application program comprises an object based on a GUI of the handheld terminal.

55.   (Currently Amended)   A method for parsing a web-document supplied from a web-server, the web-document being composed of a predetermined markup language, the method comprising ~~the steps of~~:

~~(a)~~ reading a token from the web-document by referring to a token table, extracting and separating the token;

~~(b)~~ if the extracted and separated token is not a defined start tag or if the token is a comment or a space, ignoring the token;

~~(c)~~ when the extracted and separated token is recognized as the defined start tag, parsing an attribute of an element from the token and storing GUI-related information of the element;

14

~~(d)~~parsing contents of the element after parsing the attribute of the element;

~~(e)~~as the result of the ~~step (d)~~ contents parsing, if the contents of the element are

parsed character data, storing GUI-related information of the contents, and if the contents of the

element are not the parsed character data, determining whether an end tag appears;

~~(f)~~as the result of the previous step~~(e)~~, if the end tag does not appear, repeating

~~from~~the ~~step (a)~~ above steps, and if the end tag appears, determining whether the end tag

corresponds to the defined start tag; and

~~(h)~~as the result of the previous step~~(f)~~, if the end tag corresponds to the defined

start tag, terminating, and otherwise, ignoring and returning.

56.   (Currently Amended) The method of claim 55, wherein the ~~step (e)~~ attribute

parsing comprises ~~the steps of~~:

if the extracted and separated token does not include a defined start tag, reading

the data continuously until the end tag appears, thereby ignoring the token; and

reading a new token.

57.   (Currently Amended) A handheld terminal, comprising:

an integral parser ~~for parsing~~ that parses web-documents composed of a plurality

of predetermined markup languages on the basis of elements, wherein the integral parser

includes a word parser that extracts and separates tokens of the web-document supplied to the

terminal regardless of a kind of a markup language used to compose the web-document by

referring to the token table, and wherein the word parser includes an attribute parser configured

to recognize at least one of a name of an attribute or a value of the attribute;

a memory ~~for storing~~ that stores information parsed by the integral parser; and

an application program using information extracted from the integral parser.

58.   (Currently Amended) A system, comprising:

a content provider configured to provide first type documents using a first

markup language and second type documents using a second markup language different from

the first markup language; and

a handheld terminal that receives the first and second type documents from the

content provider, wherein the handheld terminal comprises~~;~~:

an integral parser configured to parse both a first type document and a

second type document on the basis of elements to extract information thereof, wherein the

integral parser includes a word parser that extracts and separates tokens of the web-document

supplied to the terminal regardless of a kind of a markup language used to compose the web-

document by referring to the token table, and wherein the word parser includes an attribute

parser configured to recognize at least one of a name of an attribute or a value of the attribute;

and

an application program configured to receive the information extracted from the integral parser.

59. (New) The system of claim 51, wherein the attribute parser includes a first attribute parser configured to recognize a name of an attribute and a second attribute parser configured to recognize a value of the attribute.

60. (New) The system of claim 59, wherein if the value of the attribute is a keyword, the first attribute parser recognizes the name and the value of the attribute at once without distinguishing the name from the value.

61. (New) The system of claim 30, wherein the different tokens are as follows: <html> represents an element type, <p>html</p> represents parsed character data, <!--html--> represents a comment.